Software-Defined Network: A view from Summer Joint Techs focus-day

Inder Monga

August 14th, 2012

Chief Technologist

Energy Sciences Network,

Scientific Networking Division, LBNL

Context

Software-defined networking (SDN) focus-day at Summer Joint Techs

- Stanford, CA: Tuesday, July 17th, 2012
  - http://events.internet2.edu/2012/jt-stanford/agenda.cfm
- Co-chairs: Matt Davy and Inder Monga

Engaging mix of speakers from inventors, visionaries, researchers and industry leaders

Grant's challenge – 'an update on state of SDN, a JT perspective'

- Much tougher to bring together varying ideas from a rapidly evolving field

This talk:

- Subjective
- Highlights
- Policy-free (hard to do, so not completely)
- Content leveraged and acknowledged, apologies in advance for omissions

Agenda

Introduction

- Brief History of OpenFlow
- Software-defined networking

SDN Components

Use Cases

- Vendor
- Campus
- Service-provider

Research

Looking forward

Brief History of OpenFlow

The beginning: SIGCOMM 2007

Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
Stanford University

Scott Shenker
U.C. Berkeley and ICSI

Main features:

- Single Controller
- Network-wide fine-grain policy
- Admittance and routing of flows (network-wide)

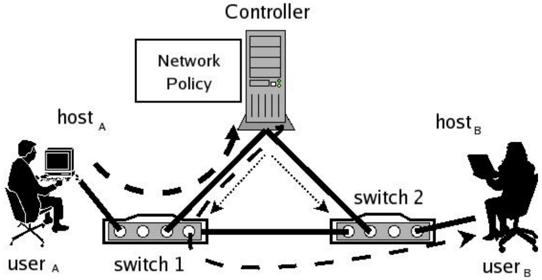
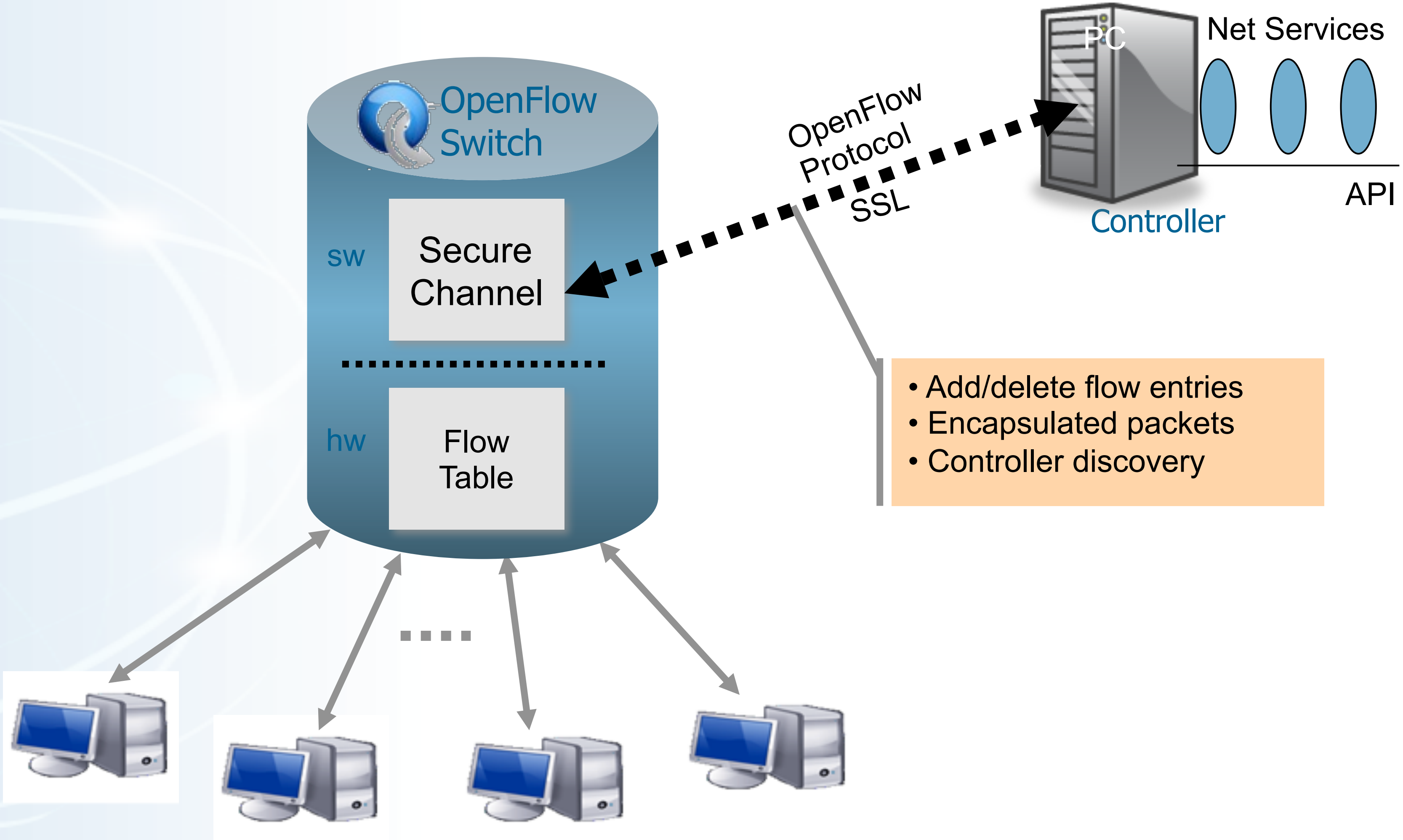


Figure 1: Example of communication on an Ethane network. Route setup shown by dotted lines; the path taken by the first packet of a flow shown by dashed lines.

OpenFlow 2010



Slide courtesy Guru Parulkar

OpenFlow v1.0
Flow table entry

Rule	Action	Stats

Packet + byte counters

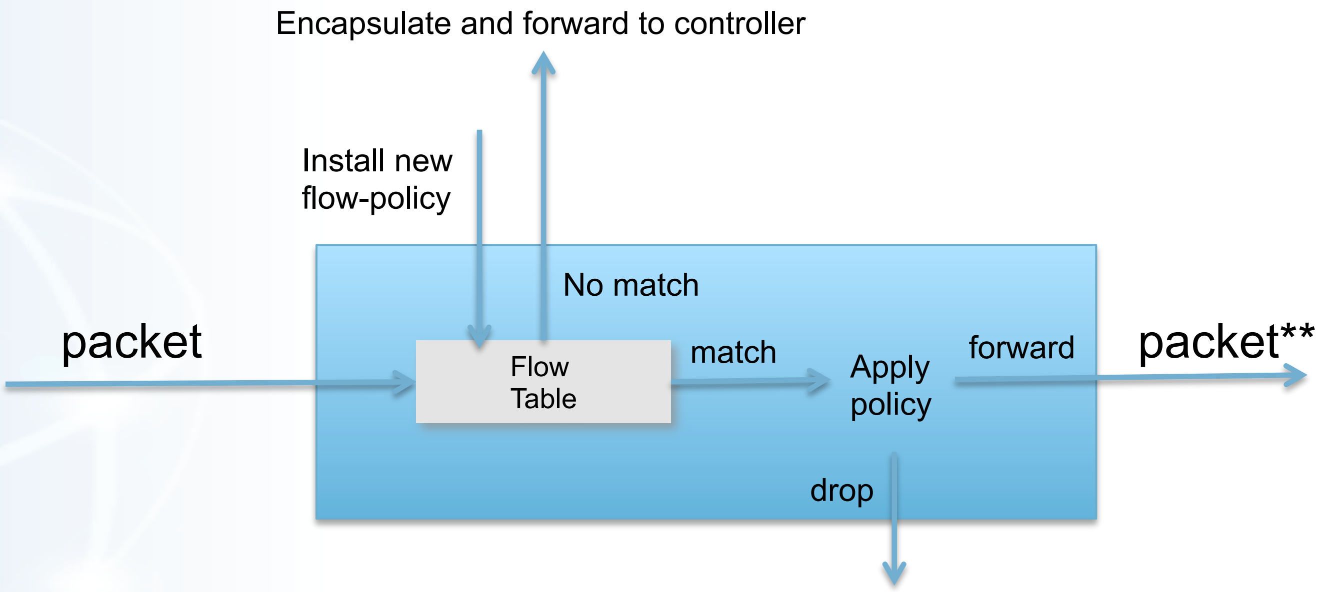
1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport

+ mask

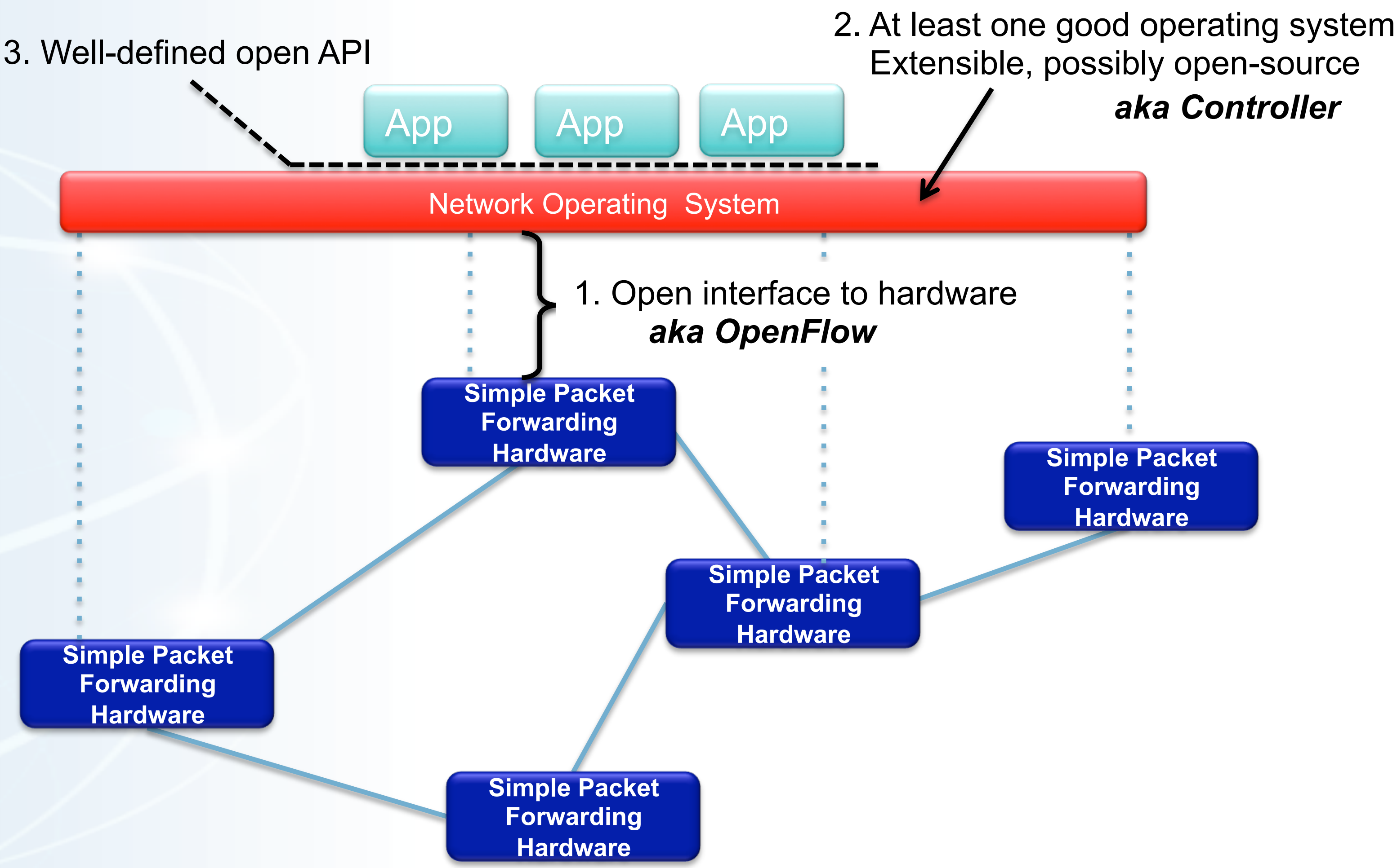
Slide courtesy Guru Parulkar

OpenFlow v1.0
Flow-table in action



Generic piece of networking hardware (virtual or real)

Software-defined networking: First instantiation of the vision



Base slide courtesy Nick McKeown/Guru Parulkar, modified for use

Software-Defined Networking (SDN): A definition

Martin Casado (Nicira founder, keynote JT)

- Programmatic interface to the forwarding plane

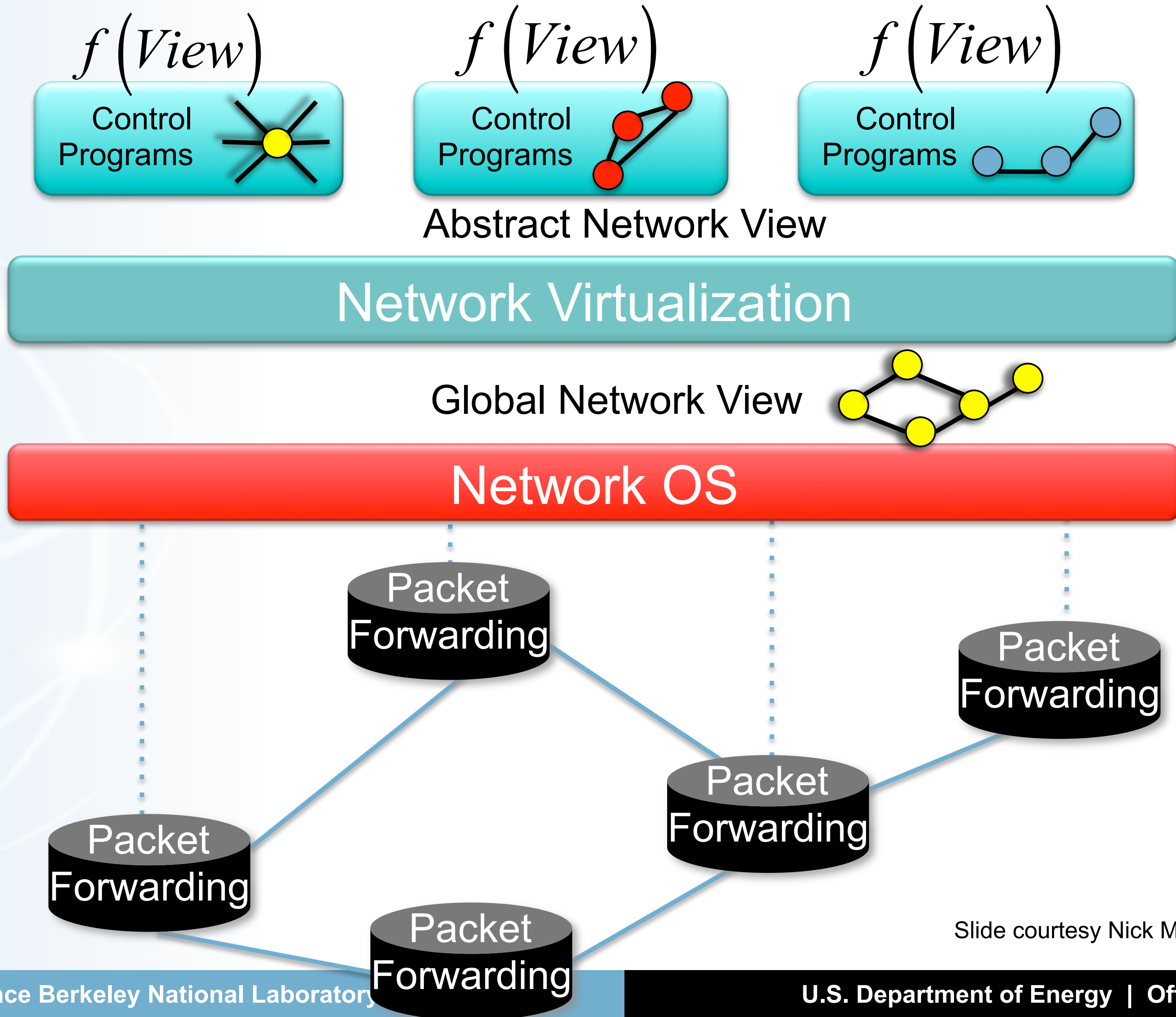
- Generalized forwarding model to allow evolution through software changes [and hardware]

- Decouple the [state] distribution model of the data plane and control plane

[] = items in square brackets are my interpretations.

http://events.internet2.edu/2012/jt-stanford/agenda.cfm?go=stream&sessid=10002431&streamtype=21&live=0

ONS 2012: Software Defined Network (SDN)

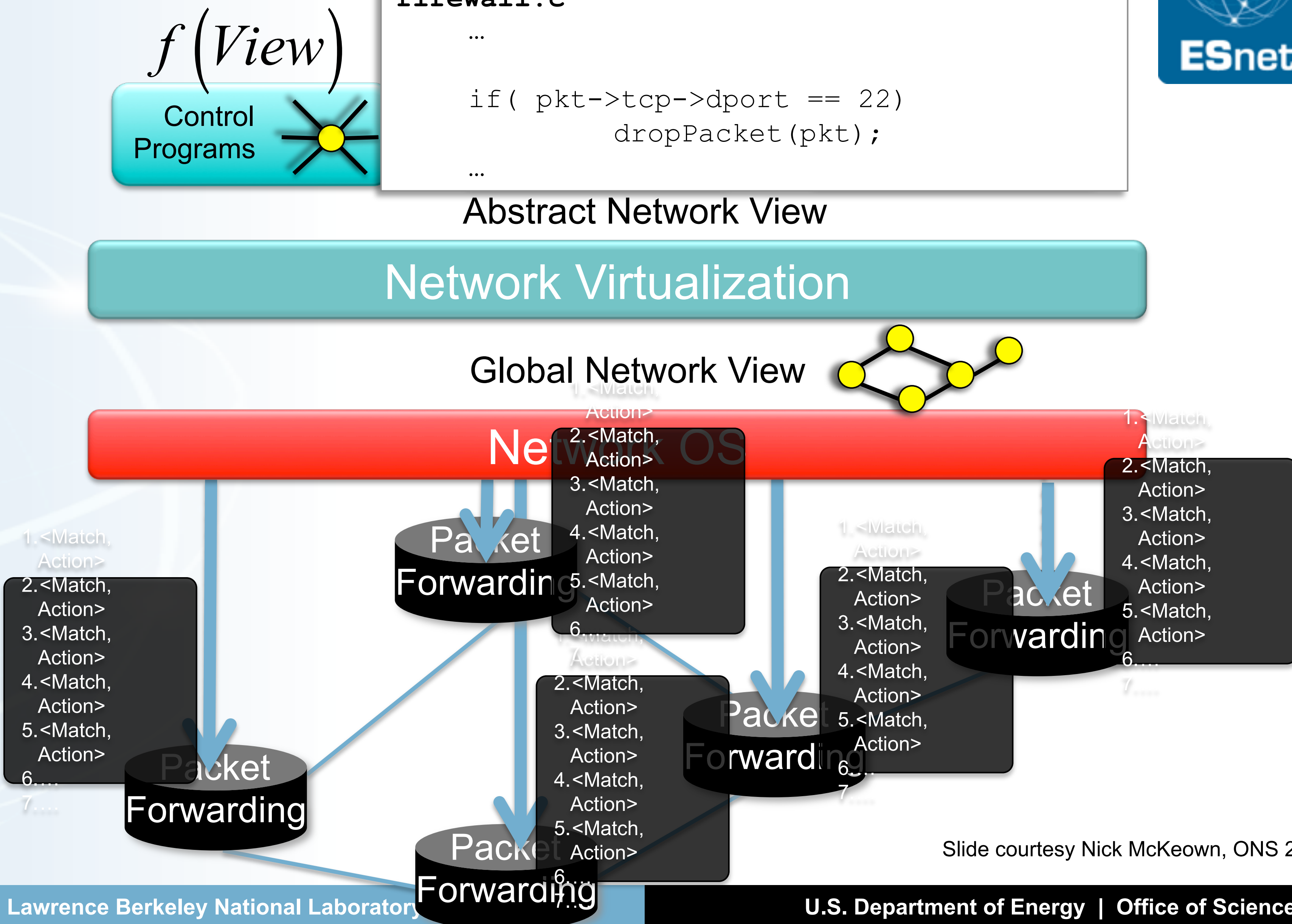


Slide courtesy Nick McKeown, ONS 2012

ONS 2012: Software Defined Network (SDN)

```
firewall.c
    …

    if( pkt->tcp->dport == 22)
            dropPacket(pkt);
    …
```



Slide courtesy Nick McKeown, ONS 2012

Agenda

Introduction

- Brief History of OpenFlow
- Software-defined networking

SDN Components

Use Cases

- Industry
- Service-provider

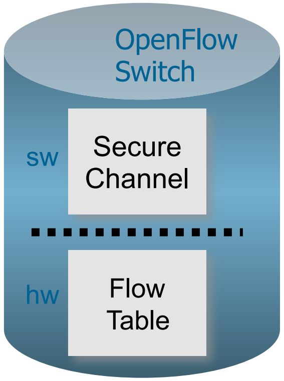
Research and Experimentation

Looking forward

SDN Components: Network

Major network vendors have announced product support for OpenFlow

- HP, NEC, Brocade, Cisco….
- Others are working on something in-house, and have beta available



Most support is of OpenFlow 1.0

What does this mean?

- OpenFlow protocol supported in hardware
- Allows programmability of the flow table through the protocol
- Partitioning of the switch, the flow-table size, etc. vary by vendor

Firmware updates likely as OF standards and controllers evolve

SDN Components
Controller

An critical part of the solution

- provides the programmability, reliability, application interface etc.
- Needs to be as reliable/redundant as the hardware layer

Hot area for development, many open-source controllers

- NOX, POX, Trema, SNAC, Flowvisor, Beacon, Floodlight

Commercial controllers are now available as well and helped with open-source controllers

- NEC ProgrammableFlow
- BigSwitch (Floodlight)
- Nicira (NOX now unsupported)
- Cisco

SDN Components
Controller (contd.)

Biggest area for innovation

- Network vendors typically haven't built single logically centralized software stack that scales immensely
- Requires skills and tools from the massively scalable application world

Short preview: Floodlight open-source controller from BigSwitch

Floodlight Overview

An Apache licensed OpenFlow Controller

- Developer friendly Apache license
- Easy to use, extensible Java development environment
- Enterprise grade - Core engine used and supported by Big Switch Networks (running in production today)
- Supports a broad range of physical and virtual OpenFlow switches
- OF 1.0 compliant today – future OF versions on the way

Courtsey Mike Cohen, http://events.internet2.edu/2012/jt-stanford/agenda.cfm?go=session&id=10002467&event=1232

Floodlight Users and Contributors

Floodlight Adopters:

- University research
- Networking vendors
- Users
- Developers / startups

Courtesy Mike Cohen, http://events.internet2.edu/2012/jt-stanford/

Floodlight Programming Model

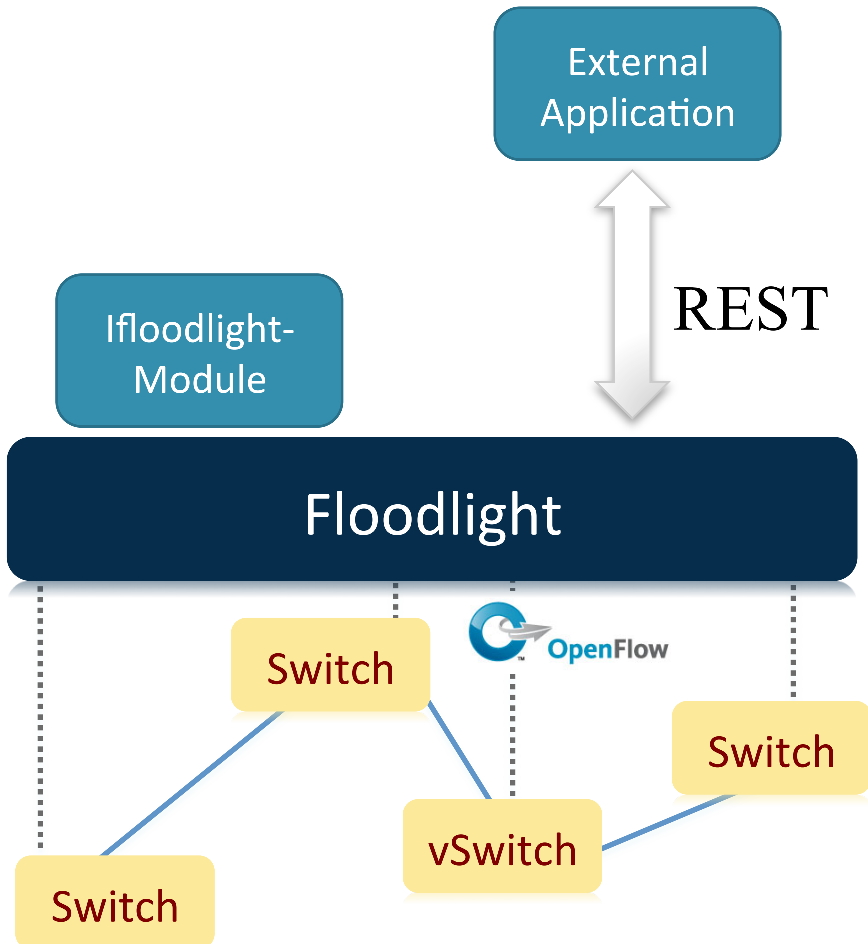
Northbound APIs

IFloodlightModule

- Java module that runs as part of Floodlight
- Consumes services and events exported by other modules
  - OpenFlow (ie. Packet-in)
  - Switch add / remove
  - Device add /remove / move
  - Link discovery

External Application

- Communicates with Floodlight via REST
  - Virtual networks
  - Normalized network state
  - Static flows



Courtesy Mike Cohen, http://events.internet2.edu/2012/jt-stanford/

Floodlight OpenStack Integration

big switch networks

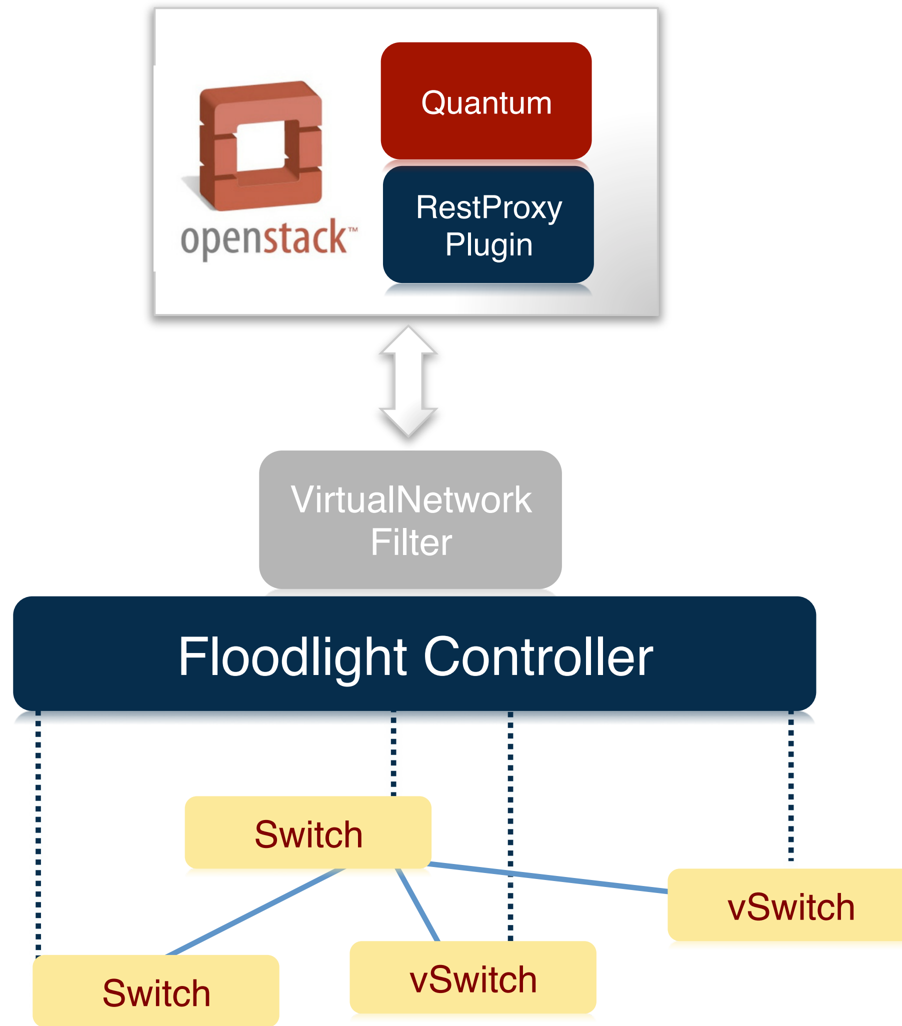
Virtual Networking Support

Components:

- RestProxy plugin runs inside Quantum module in OpenStack
- VirtualNetworkFilter implements layer 2 isolation based on MAC

Highlights:

- Supports physical and virtual switches in OpenFlow networks
- Caveats:
  - No multicast and broadcast isolation
  - All DHCP traffic allowed



Courtsey Mike Cohen, http://events.internet2.edu/2012/jt-stanford/agenda.cfm?go=session&id=10002467&event=1232

Evolving the OpenFlow and SDN components

Open Networking Foundation (ONF)

- Standards Organization, SDN focused
- https://www.opennetworking.org/index.php

Open Networking Summit (ONS)

- OpenFlow/SDN conference
- http://opennetsummit.org/

Open Network Research Consortium (ONRC)

- Industry sponsored research center at Stanford
- “Create a solid, scientific foundation for SDN”
- http://onrc.stanford.edu/

ON.LAB

- Develop, support and deploy open-source SDN tools and platforms
- http://onlab.us/

Standards – Open Networking Forum

OpenFlow standards

Evolution path:

- OF 1.0 (03/2010): Most widely used version, MAC, IPv4, single table (from Stanford)
- OF 1.1 (02/2011): MPLS tags/tunnels, multiple tables, counters (from Stanford)
- OF 1.2 (12/2011): IPv6, extensible expression
- OF-Config 1.0 (01/2012): Basic configuration: queues, ports, controller assign
- OF 1.3.0 (04/2012): Tunnels, meters, PBB support, more IPv6
- OF-Config 1.1 (04/2012): Topology discovery, error handling
- OF-Test 1.0 (2H2012): Interoperability & conformance test processes, suites, labs

Goals:

- Widespread adoption, experimentation w/OF 1.3.x
- Accommodate current merchant silicon while moving beyond current limits
- Use market feedback to drive future development

Slide courtesy Dan Pitt @ JT

Technical activities

Chartered Working Groups

- Extensibility (chair: Jean Tourrilhes, HP): OpenFlow protocol evolution
- Config-mgmt (chair: Deepak Bansal, Microsoft): basic switch configuration; OAM?
- Testing-interop (chair: Michael Haugh, Ixia): conformance, interop., benchmarking
- Hybrid (chair: Jan Medved, Cisco): mixed OpenFlow/legacy switches & networks

Discussion Groups

- OpenFlow-Future: forwarding-plane models
- NorthboundAPI: how the network relates to the applications (incl. OSS, BSS)
- NewTransport: OpenFlow for optical, circuits, wireless
- Market Education (chair: Isabelle Guis, Big Switch): marketing, customer value

7 Board companies, 65 others

- Urs Hölzle (Sr. VP, Engineering, Google), chairman
- Najam Ahmad (Director, Network Engineering, Facebook)
- Adam Bechtel (VP, Infrastructure Group, Yahoo)
- Stuart Elby (VP, Network Architecture, Verizon)
- Axel Clauberg (VP, IP & Optical, Deutsche Telekom)
- Yukio Ito (Sr. VP, Services & Infrastructure, NTT Communications
- Clyde Rodriguez (GM, Windows Azure Networking, Microsoft)
- Nick McKeown (Professor, EE and CS, Stanford)
- Scott Shenker (Professor, EECS, UC Berkeley and ICSI)

- A10 Networks
- ADVA Optical
- Alcatel-Lucent
- Aricent
- Argela/Turk Telekom
- Big Switch Networks
- Broadcom
- Brocade
- Ciena
- Cisco
- Citrix
- Colt
- CompTIA
- Cyan Optics
- Dell/Force10
- Elbrys
- Ericsson
- ETRI
- Extreme Networks
- EZchip
- F5 Networks
- Freescale Semi
- Fujitsu
- Gigamon
- Goldman Sachs
- Hitachi
- HP
- Huawei
- IBM
- Infinera
- Infoblox
- Intel
- IP Infusion
- Ixia
- Juniper Networks
- Korea Telecom
- LineRate Systems
- LSI
- Luxoft
- Marvell
- Mellanox
- Metaswitch Networks
- Midokura
- NCL Comms K.K.
- NEC
- Netgear
- Netronome
- Nicira Networks
- Nokia Siemens Netw.
- NoviFlow
- Oracle
- Orange/France Telecom
- Pica8
- Plexxi Inc.
- Radware
- Riverbed Technology
- Samsung
- SK Telecom
- Spirent
- Telecom Italia
- Tencent
- Texas Instruments
- Vello Systems
- VMware
- ZTE

Standards going beyond OpenFlow

Conclusions

ONF now the home of OpenFlow

- Take OpenFlow 1.1 to commercial strength – Job One
- Family of standards: foundation, building blocks, choices
- Protocols; configuration and management; compliance and interoperability
- Development, deployment, experience, feedback

More to SDN than OpenFlow

- SDN abstractions, object models, interactions
- Ecosystem for new features, new players, new business models

Technical standards + market education

- Market pull to drive the ecosystem

www.OpenNetworking.org

Dan.Pitt@OpenNetworking.org

Agenda

Introduction

- Brief History of OpenFlow
- Software-defined networking

SDN Components

Use Cases

- Industry
- Service-provider

Research and Experimentation

Looking forward

Use-Cases and Approach

Two different approaches to use-cases

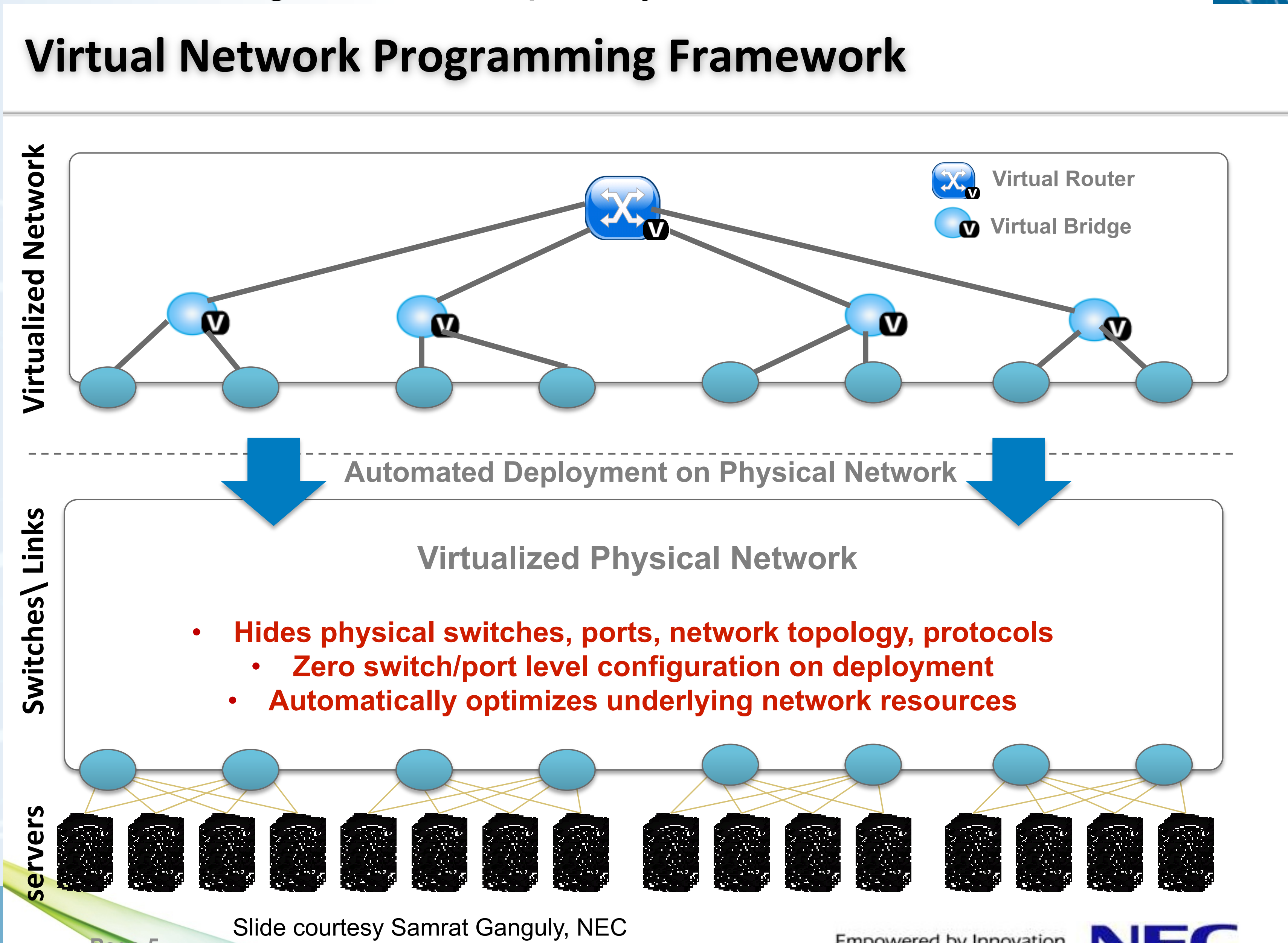
- Solve existing problems innovatively with SDN/OpenFlow concepts
  - Some vendors focused here

- Create new value-adds with SDN/OpenFlow
  - New concepts being explored

Walk-through of some use-cases to give a perspective on OpenFlow being leveraged for real-situations

NEC – Data Center

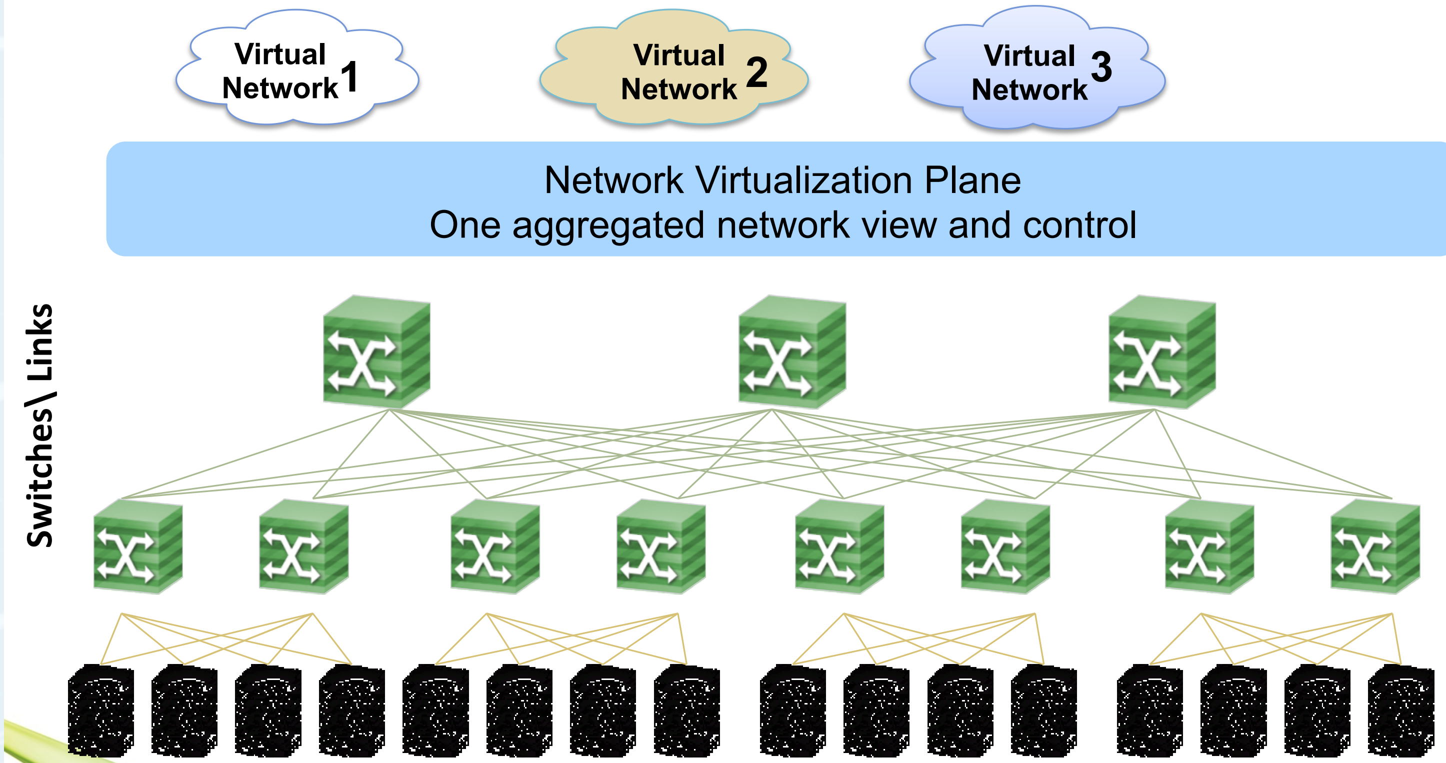
Virtualizing networks leads to simple programming: Who manages the complexity?



Slide courtesy Samrat Ganguly, NEC

Network Slicing

Create Multiple Virtual Networks



Page 6 Slide courtesy Samrat Ganguly, NEC

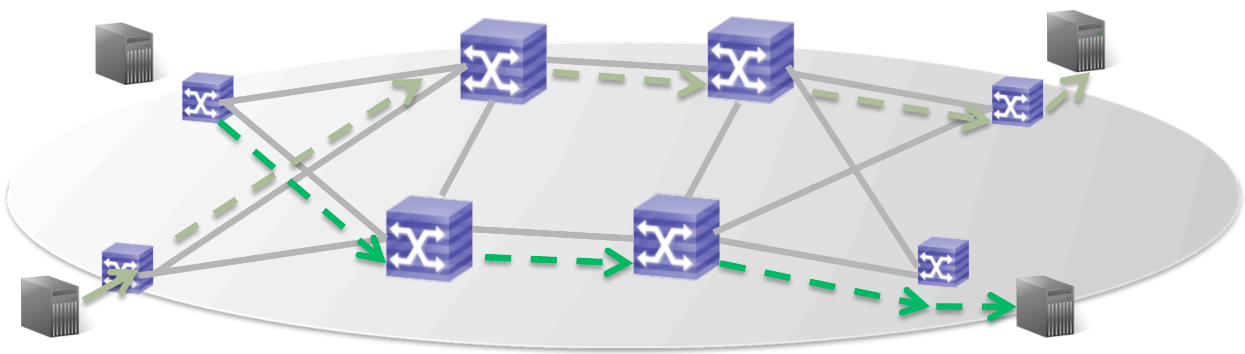
Empowered by Innovation NEC

Multipath using OpenFlow

Multipath Supporting East-West Fabric Traffic

Multipath without HW vendor lock-in
Support any interconnect topology
No complex distributed protocols

- Automatically discovers multiple paths (8-way ECMP)
- Automatically avoids loops
- No route flapping



Physical Network

 Slide courtesy Samrat Ganguly, NEC

Creating a “Science Data Network” over high-speed interfaces

Multiclass Path Policy

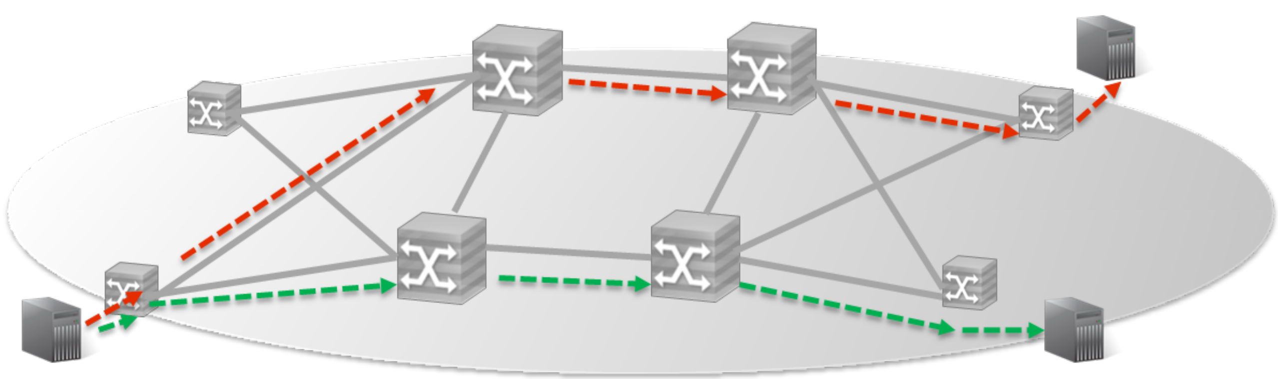
Creating Non-interfering traffic lanes for different traffic class

- Class -1 traffic: FTP and large bulk data transport
- Class -2 traffic: Messaging – latency sensitive

Use composite Packet header fields Condition to define Traffic classes

Path 1

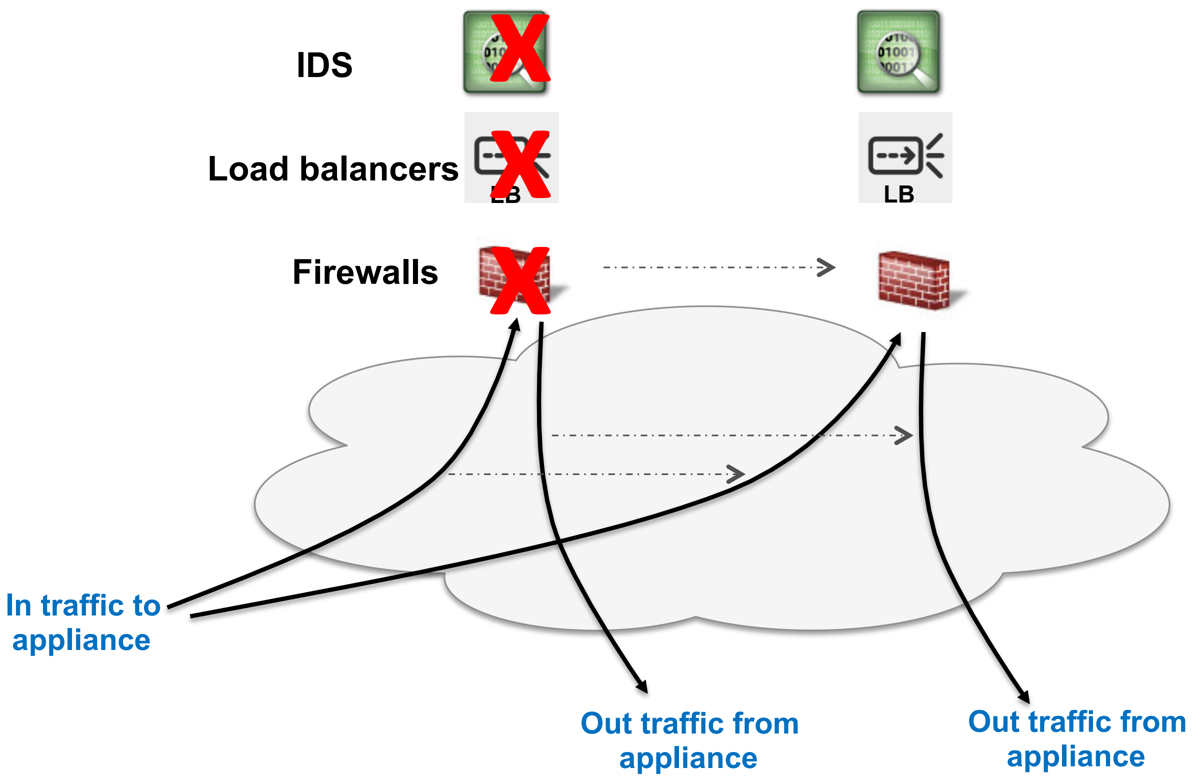
Path 2



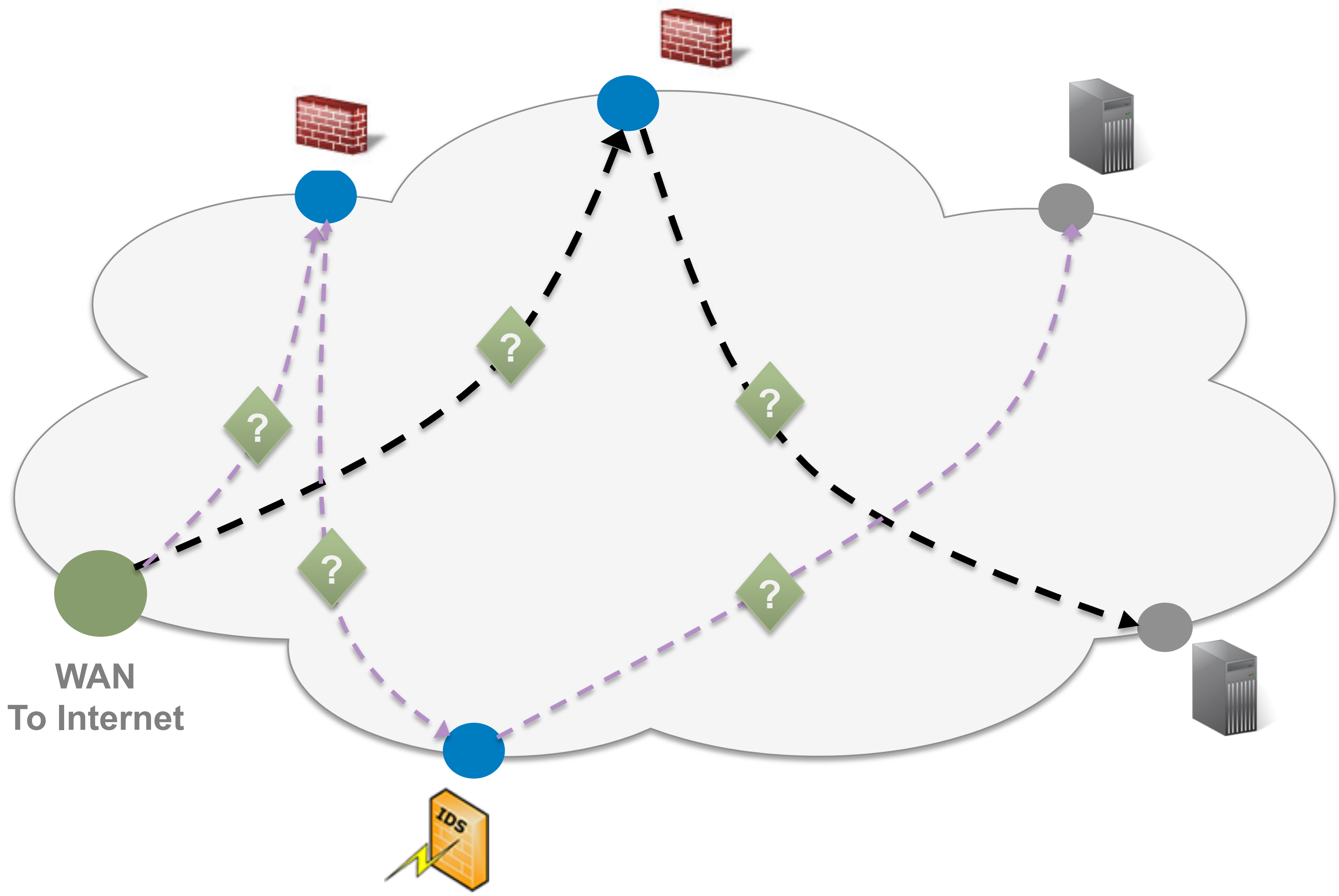
Slide courtesy Samrat Ganguly, NEC

Dynamic routing towards alternative resources

Robust Service/Appliance Failover



Dynamic Service Insertion using Conditional Routing



Slide courtesy Samrat Ganguly, NEC

Use-Cases - Cisco

Slide courtesy David Ward, Cisco

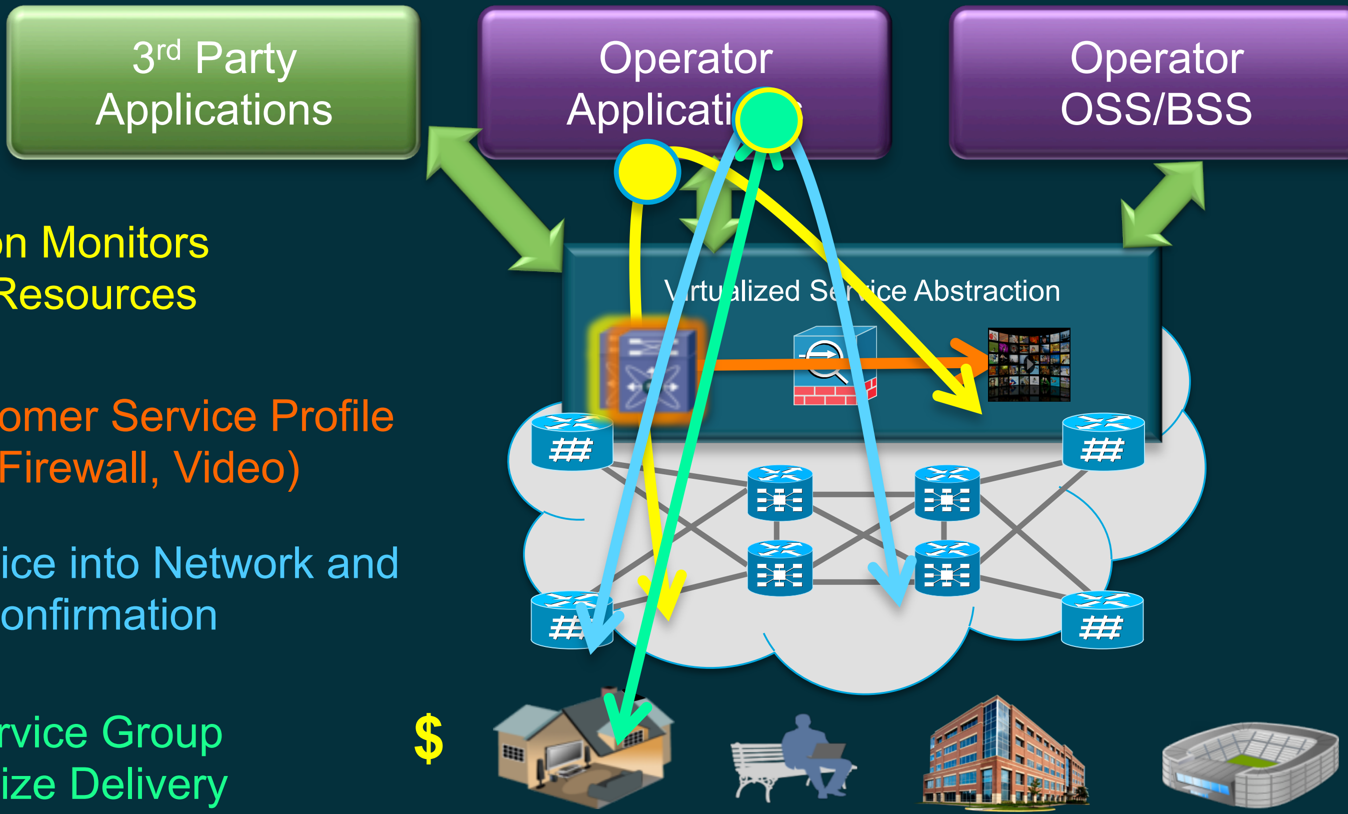
Virtualized Service Creation: Network feedback loop

1 Application Monitors Network Resources

2 Build Customer Service Profile (e.g VoIP, Firewall, Video)

3 Push Service into Network and Receive Confirmation

3 Enable Service Group and Monetize Delivery

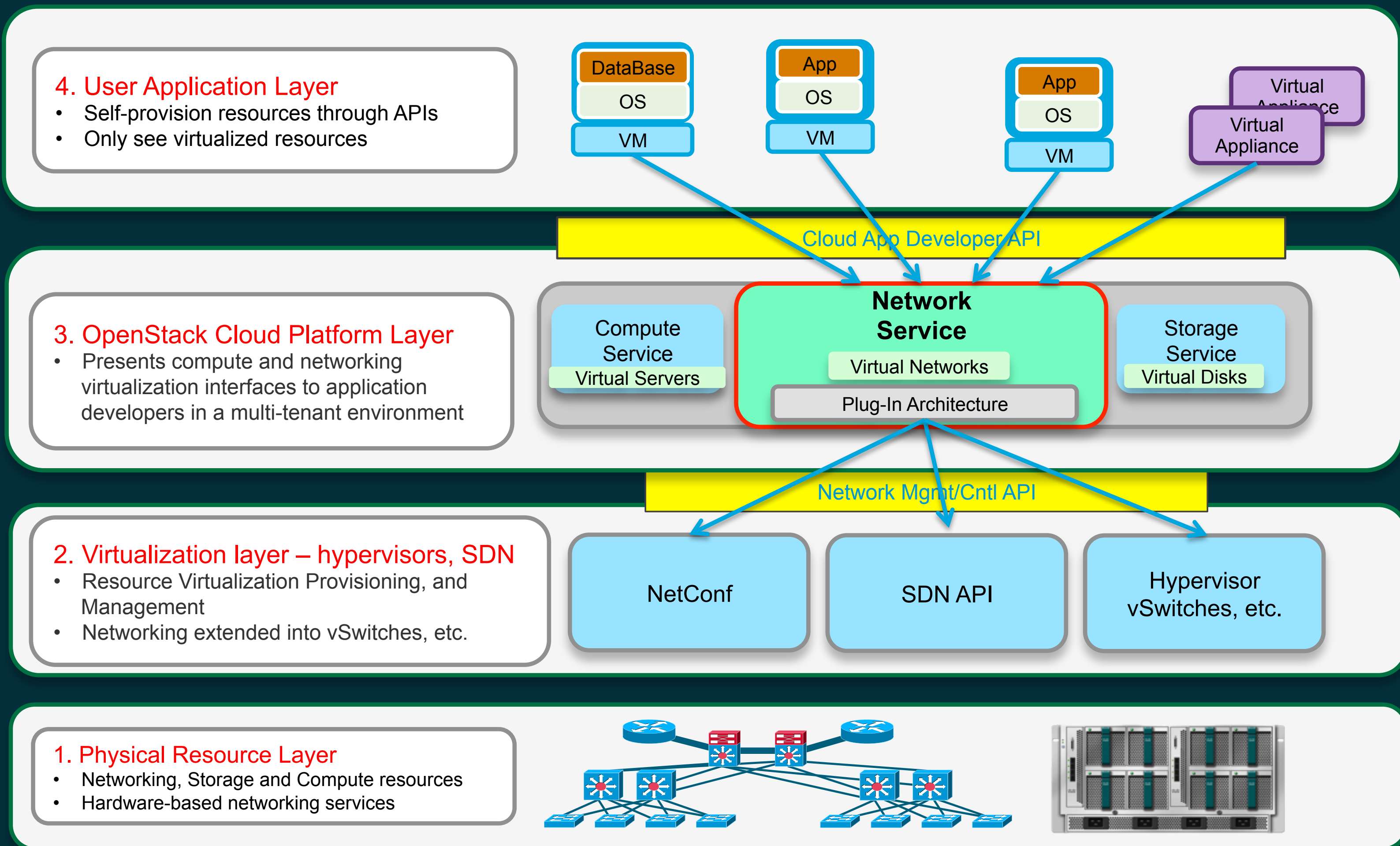


SP Opportunity: Flexibly create services based on TIGHT SLAs
Todays' Challenges: Manual provisioning, ticket entry, workflows on multiple systems
Solution: Enable agile customer service creation with service chaining
Technologies: Virtual Path, Service Chaining, Network Virtualization

Slide courtesy David Ward, Cisco

SDN and Cloud Computing

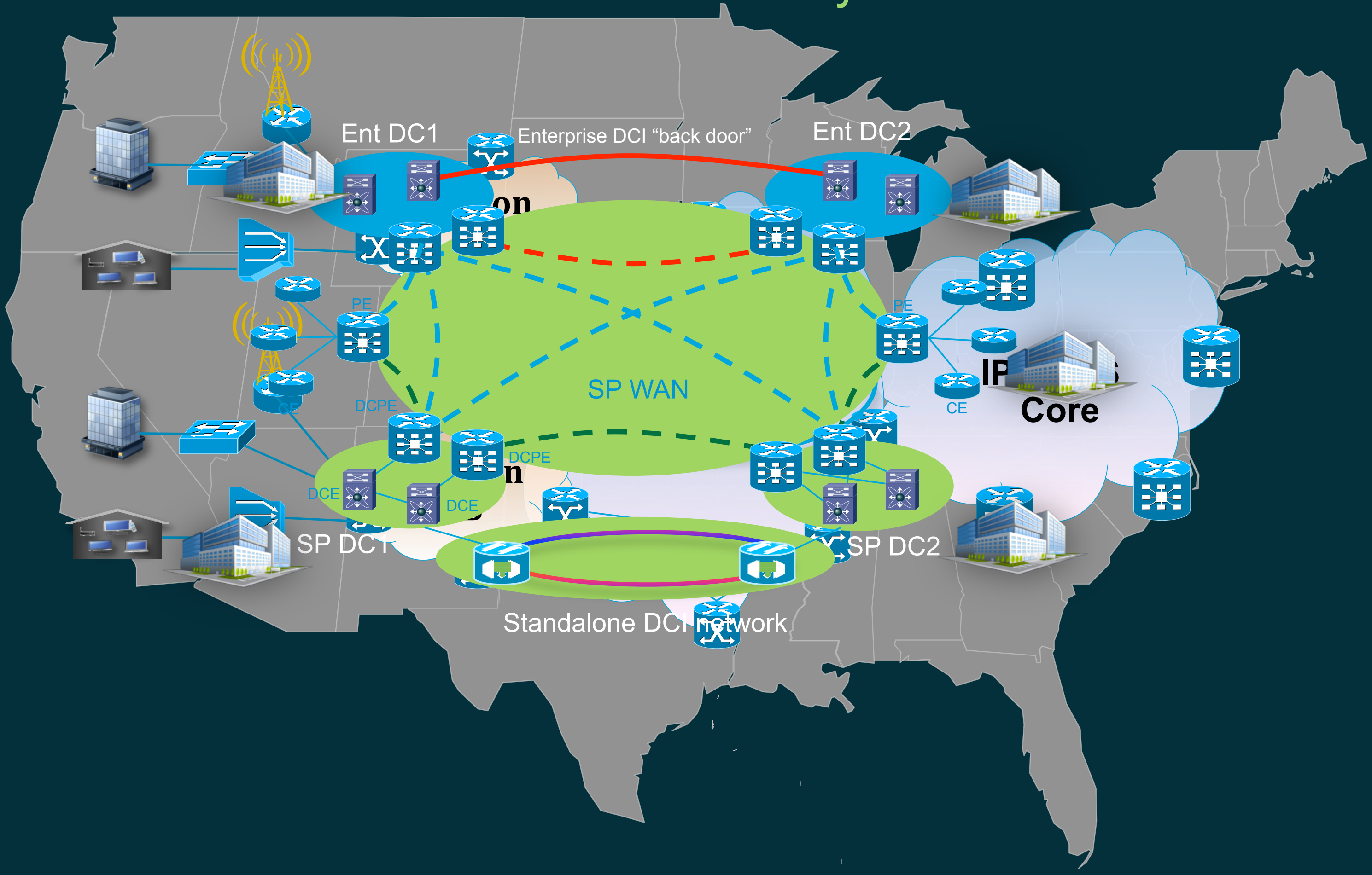
OpenStack is a platform for application developers to automate provisioning of compute, storage, and networking virtual resources



Slide courtesy David Ward, Cisco

Multi-tenant Cloud Interconnect

Just One SP WAN = At Service of Many



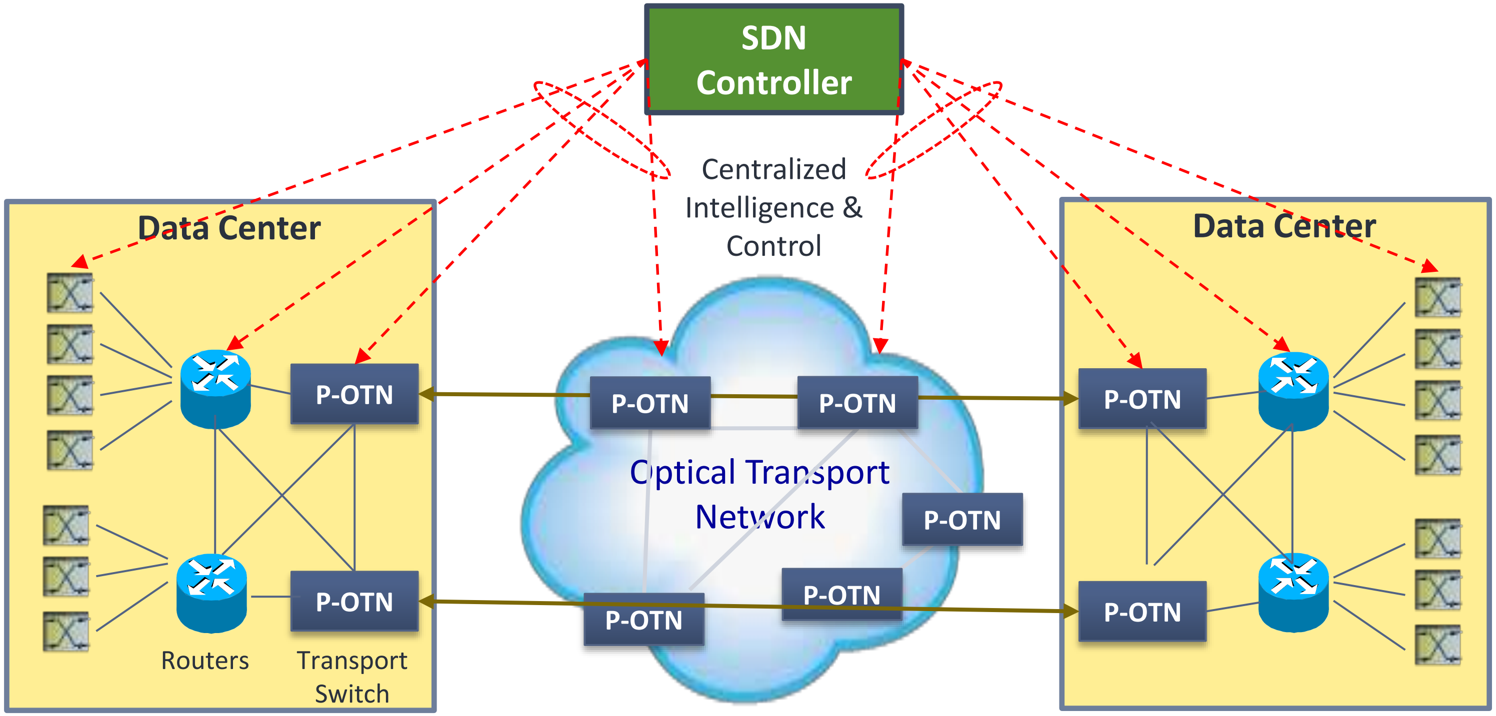
Slide courtesy David Ward, Cisco

Use-Case - Infinera

Slide courtesy Chris Liou, Infinera

SDN for Packet-Optical Transport Networks (P-OTN)

Use case example: Inter-Data Center Interconnect



- <u>SDN Early phases</u>
  - Layer 2/3 application focus
  - Provisioning only (OpenFlow)
  - No QoS
  - Fixed optical capacity model
  - Controlled trials/investigations

- <u>SDN Going Forwards…</u>
  - Layer 0/1/2/2.5/3 integration
  - Discovery, Provisioning & Monitoring
  - QoS considerations
  - Dynamic, flexible transport
  - Multi-layer resource optimization

Slide courtesy Chris Liou, Infinera

Google - WAN

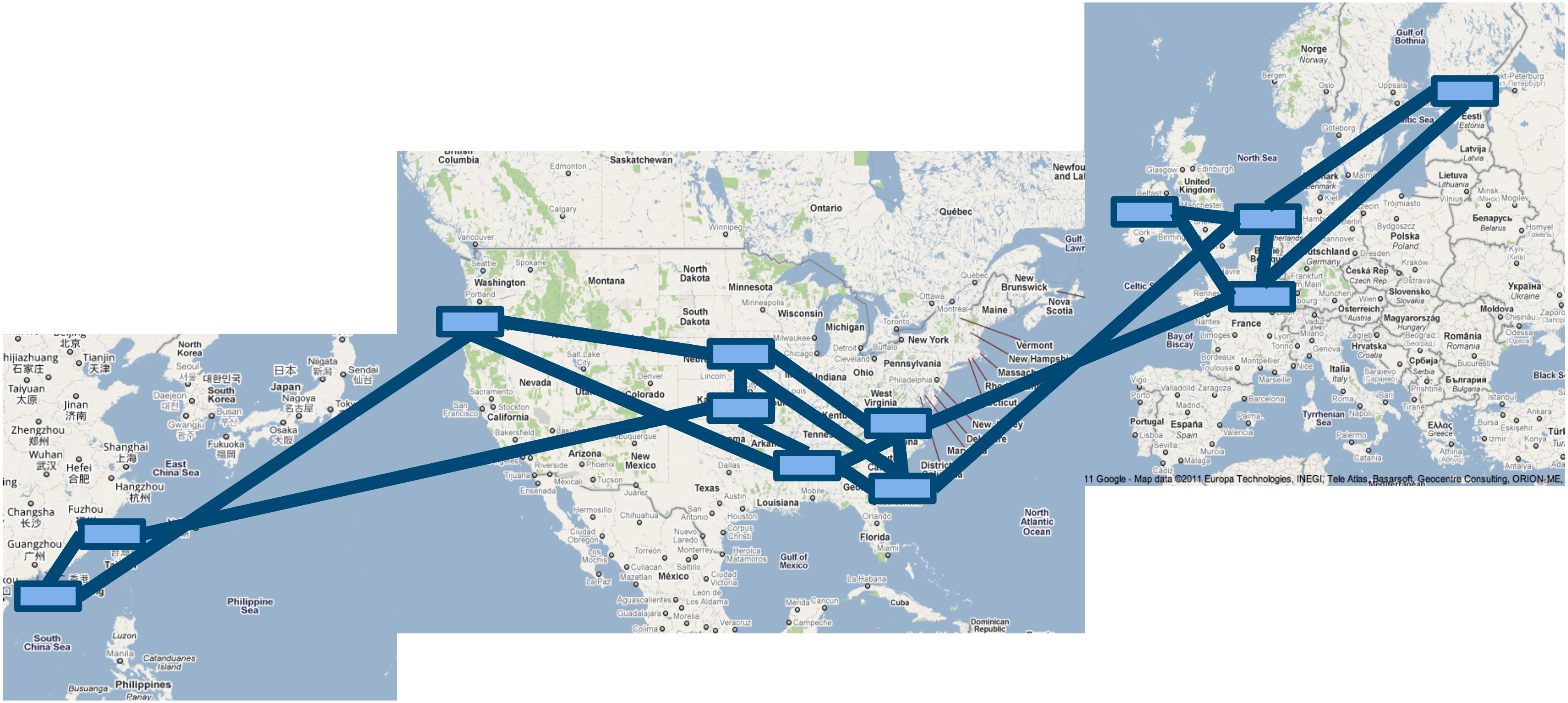
Slide courtesy Amit Agarwal, Google

Google Requirements

Deliver data

- cost effectively
- higher performance
- better fault tolerance
- manageable solution

Google's SDN powered WAN



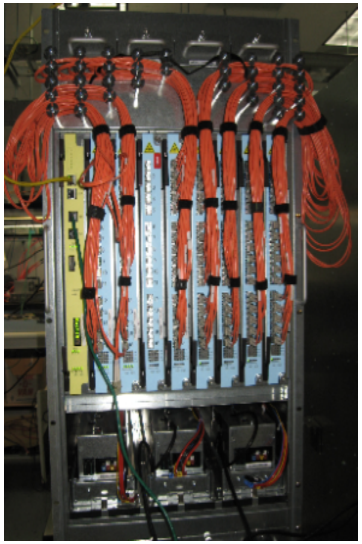
G-scale WAN - Serves traffic between datacenters

Slide courtesy Amit Agarwal, Google

G-Scale WAN Deployment

Google

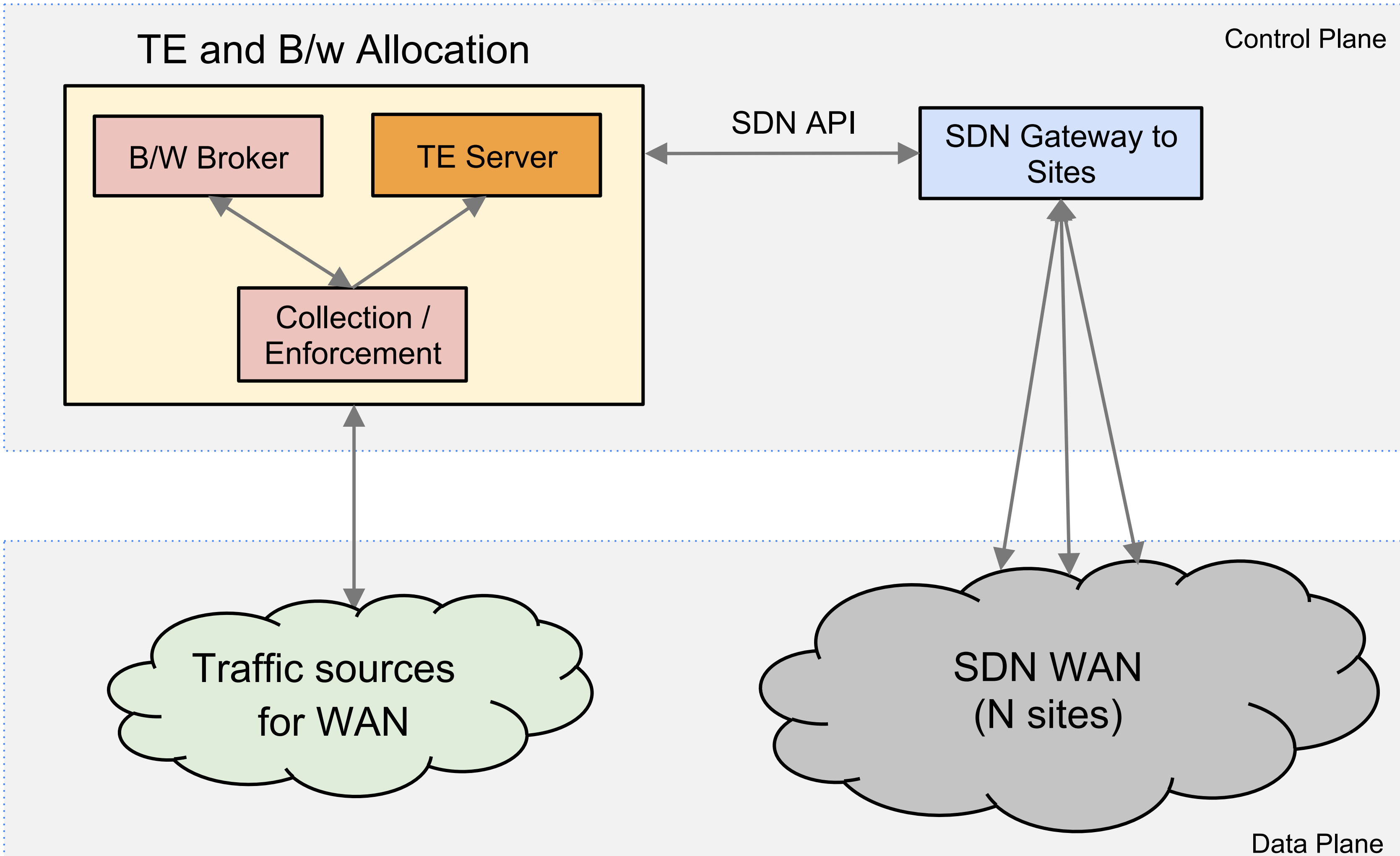
- Multiple switch chassis in each domain
  - Custom hardware running Linux
- Open source routing stacks
- OpenFlow support



Slide courtesy Amit Agarwal, Google

Centralized TE

Very similar to OSCARS architecture



Slide courtesy Amit Agarwal, Google

Benefits of SDN

- Unified view of the network fabric
  - Simplifies configuration, management and provisioning
- High utilization – up to 95% utilization of the network
- Faster failure handing
  - Systems converge faster to target optimum and behavior is predictable

Slide courtesy Amit Agarwal, Google

Benefits of SDN

Google

- Faster time to market/deployment
  - only features needed are developed and rigorous testing helps accelerate deployment
- Hitless upgrades
- High fidelity test environment
  - Emulate network in software to help in testing, verification and running “what-if” scenarios
- Elastic compute
  - use latest generation of servers

Slide courtesy Amit Agarwal, Google

Agenda

Introduction

- Brief History of OpenFlow
- Software-defined networking

SDN Components

Use Cases

- Industry
- Service-provider

Research and Experimentation

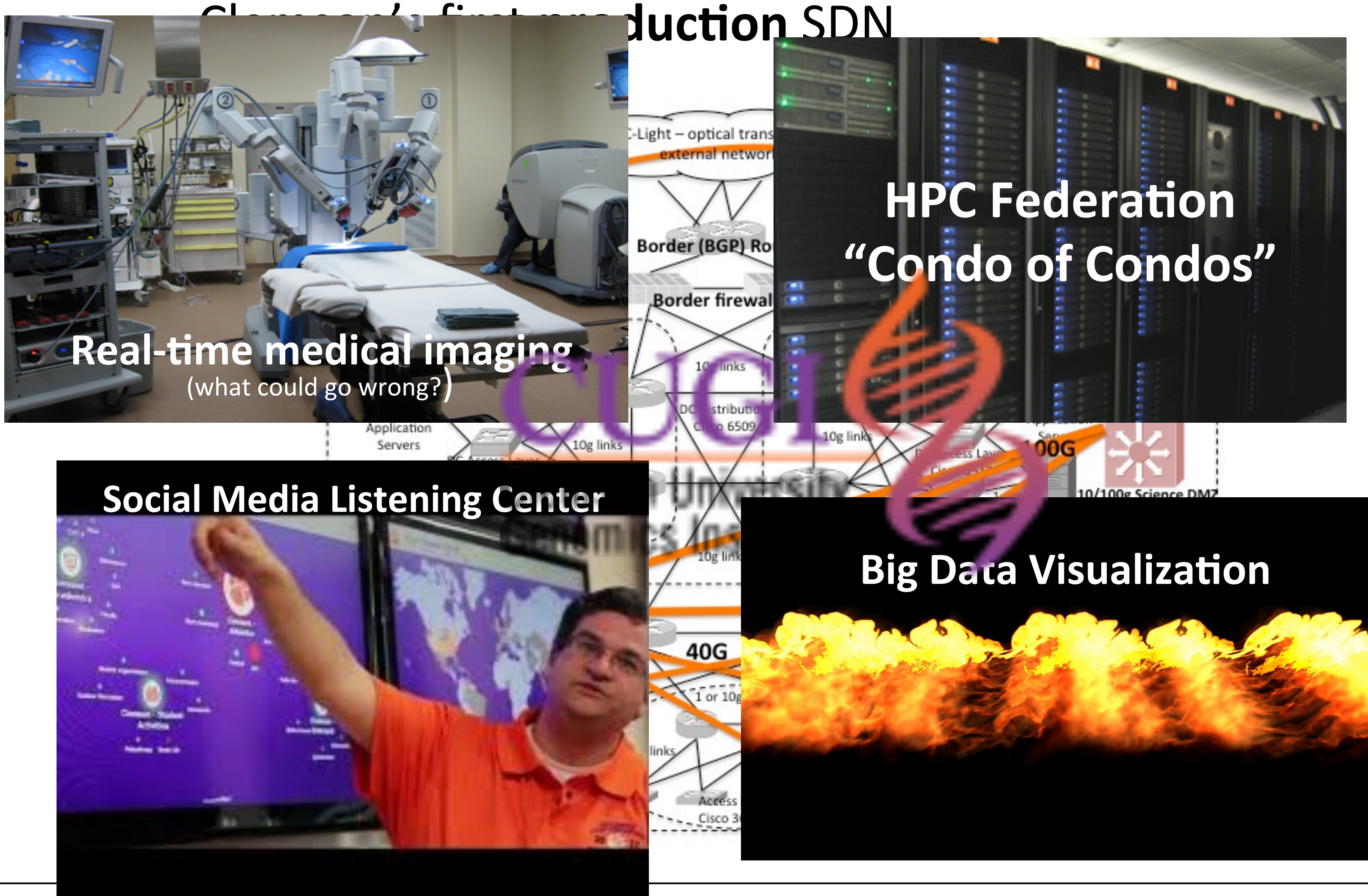
Looking forward

Clemson and ESnet: R&E Campus and WAN

Evolution of OpenFlow @ Clemson

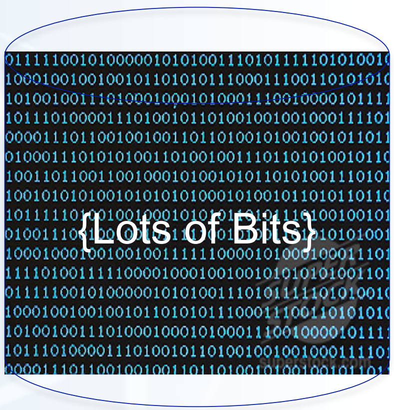
The Openflow extension for Science DMZ

duction SDN

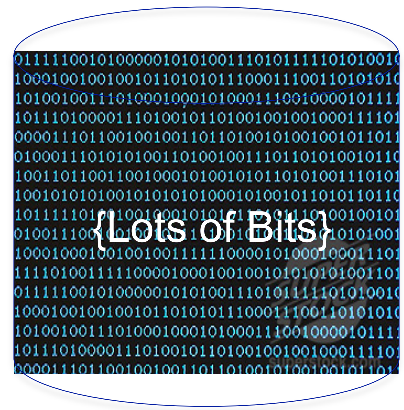


Two Prime Requirements

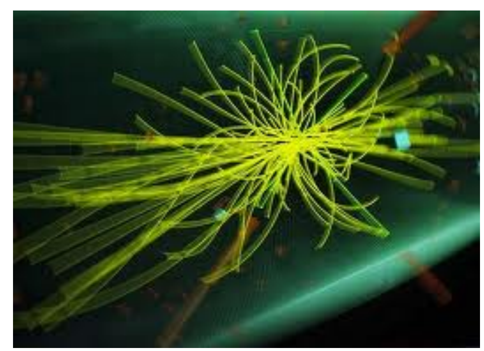
1. Data Mobility



Long latencies (RTT)
Multi-domain
Multi-vendor
Multi-technology



2. Global Collaboration
   – Higgs Boson



SDN and the Wide-Area Network

Software-Defined Networking has already been well adopted by the R&E wide-area networks

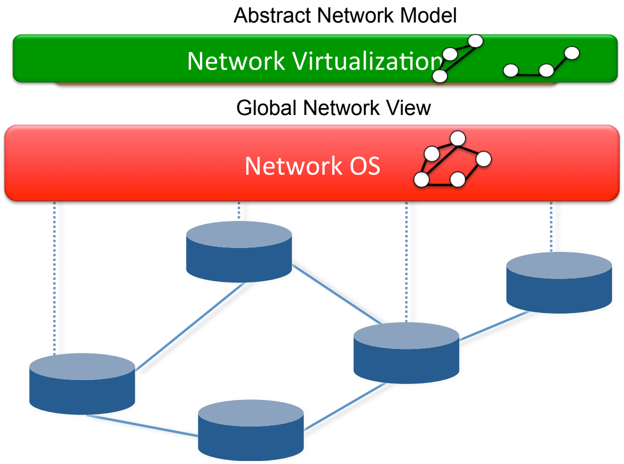
- OSCARS centralized advanced reservation and provisioning

What's different with this SDN/OF wave?

- Formal concept of a network OS
- Abstract Network model

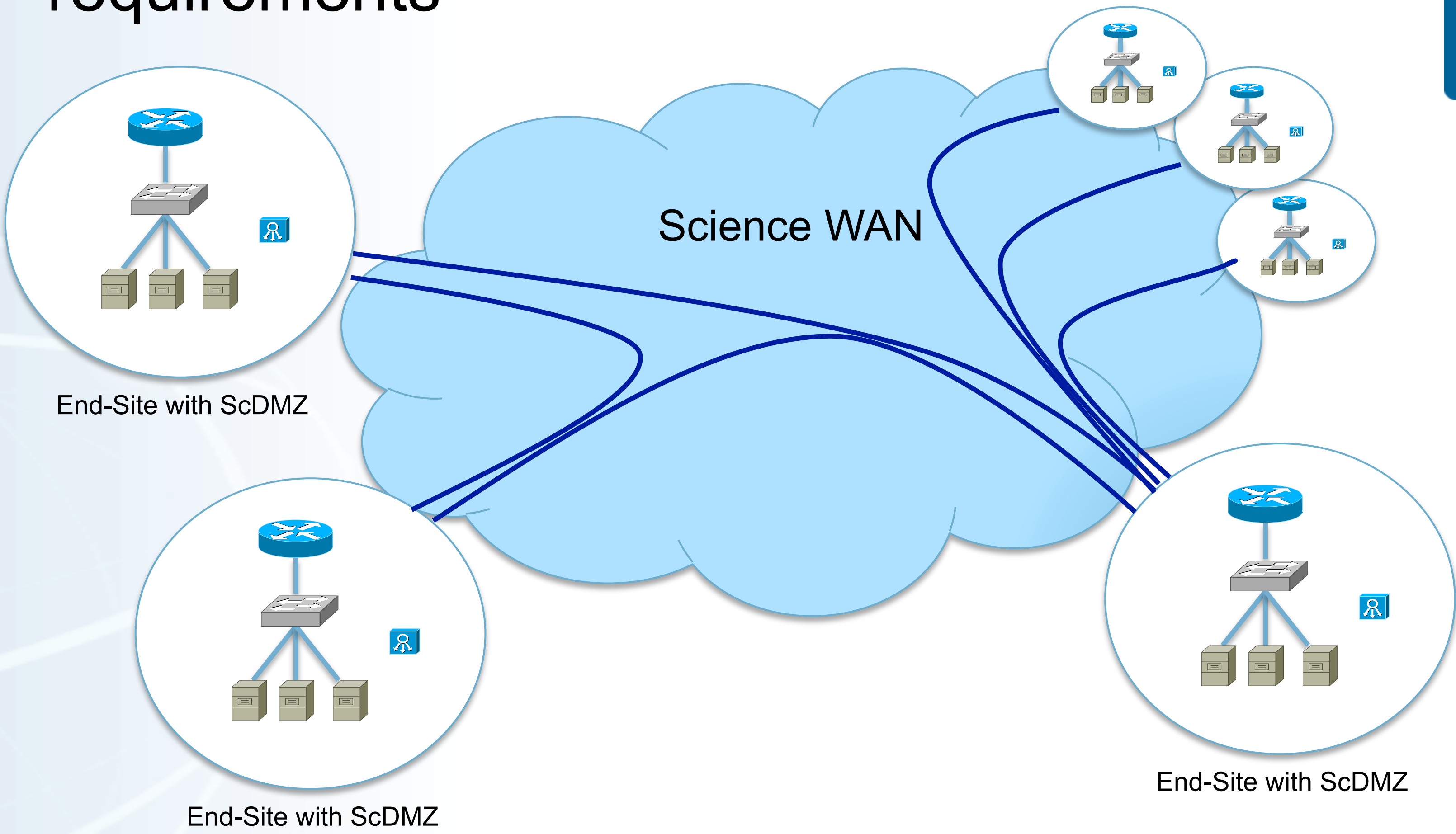
What are the fundamental network abstractions?

- All discussions on standard Northbound APIs are fruitless unless we define these



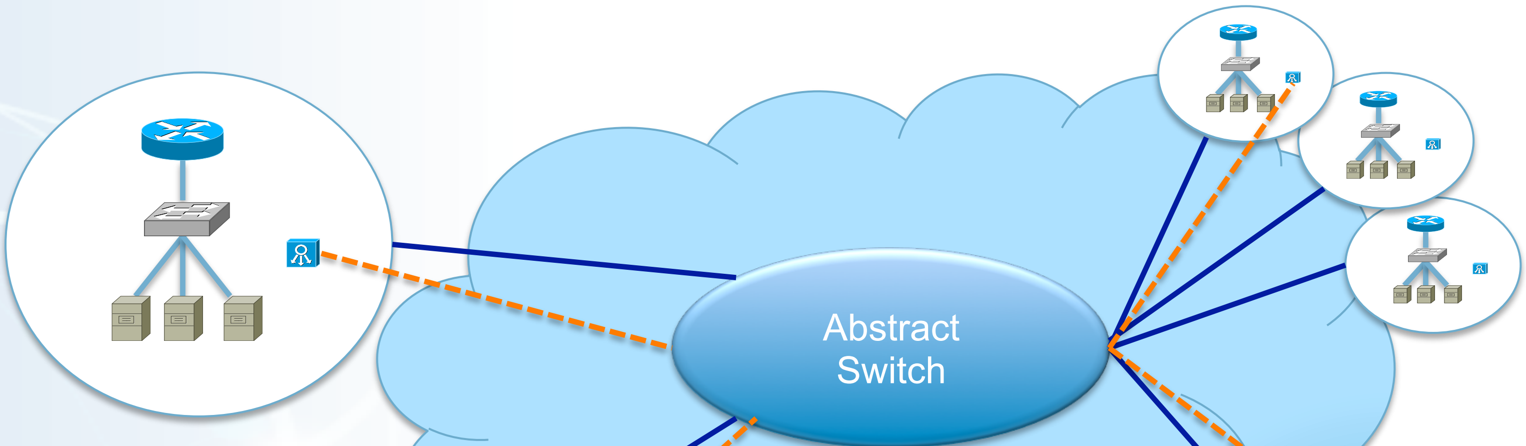
SDN with Virtualization
Guru Parulkar, from
"OpenFlow/SDN: A new approach to networking"

Going back to Science Networking requirements



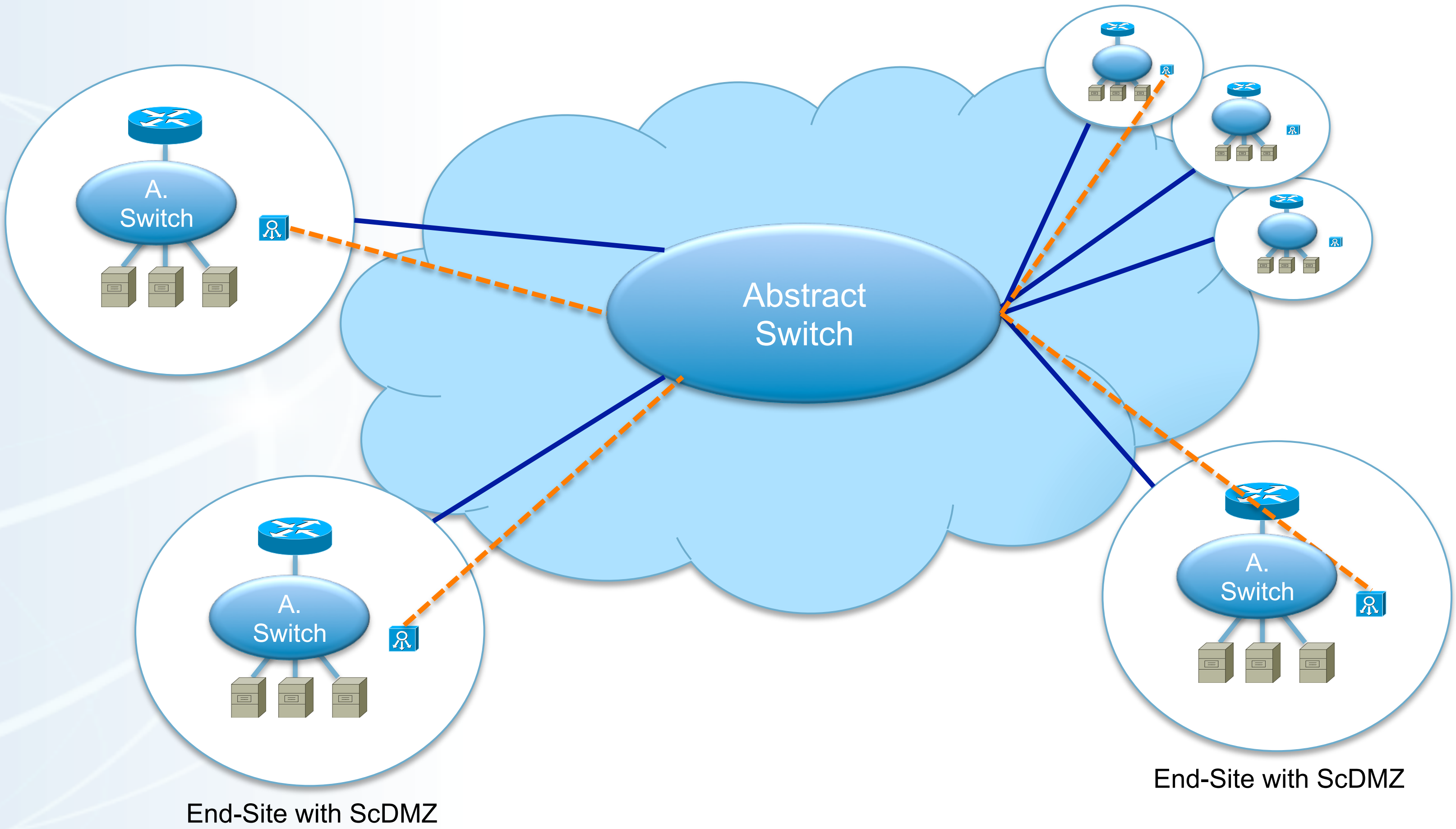
Dynamic Point to point circuits scale reasonable well,
but don't meet all global collaboration requirements (requirement #2)

A wide-area abstraction = Logical programmable OF switch



- Multi-point to multi-point connectivity
    - While leveraging the multi-domain, advanced reservation capabilities of R&E networks
- OpenFlow interface for flow programmability by the ScDMZ OF controllers
- Can be sliced further into virtual-switches or topologies

Recursive Abstraction



Practical considerations of a programmable switch abstraction

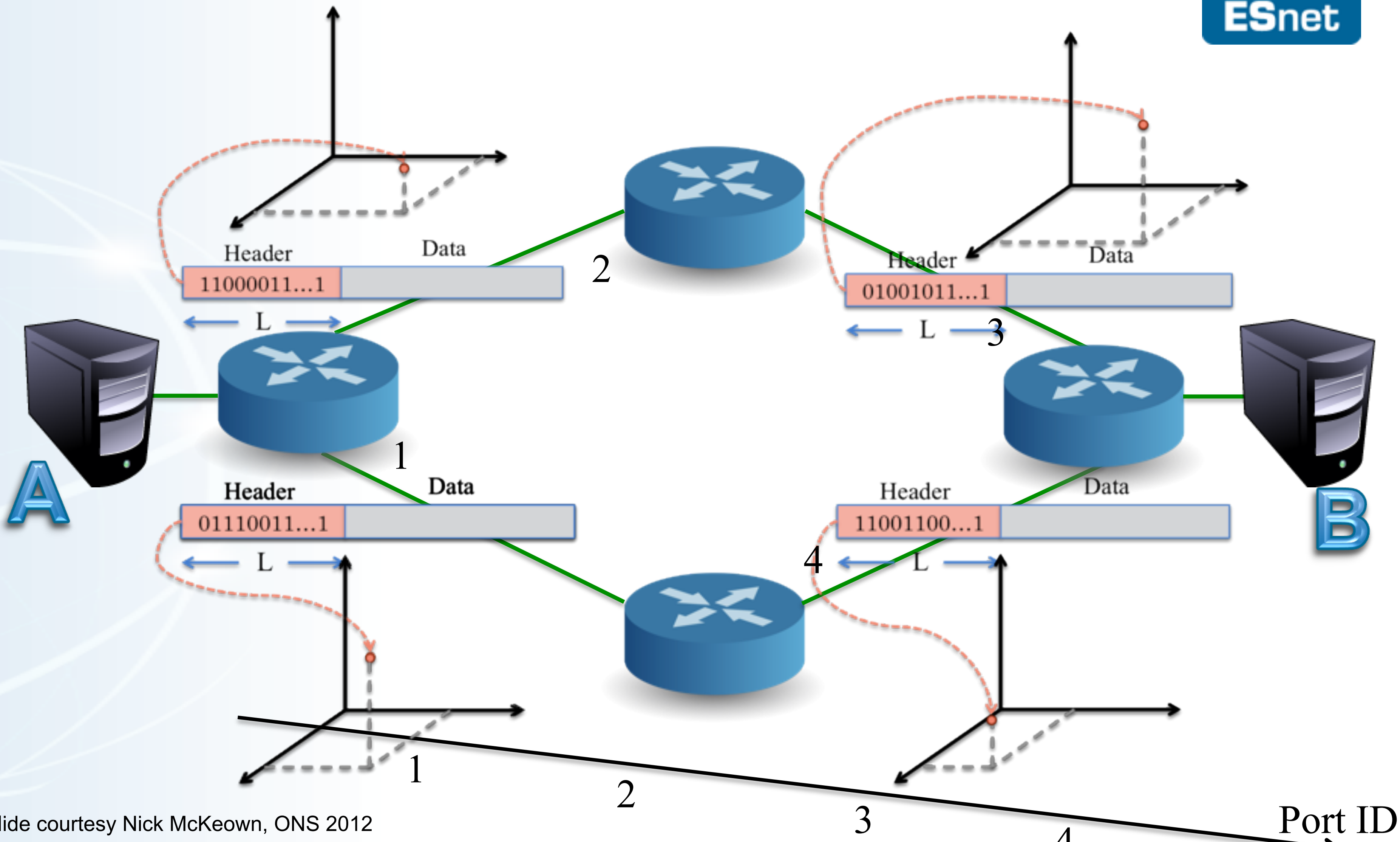
- Do not need to have all OF devices in the WAN
- Do not need to have OF support in the Site
  - Just a controller
- No new protocols or API
- Capable of supporting both L2/L3 switching
- Supports all models of end-to-end conversations aka brokers
  - ECSEL, GENI…

Header Space Anaylsis

Peyman Kazemian, James Zeng, George Varghese (UCSD)

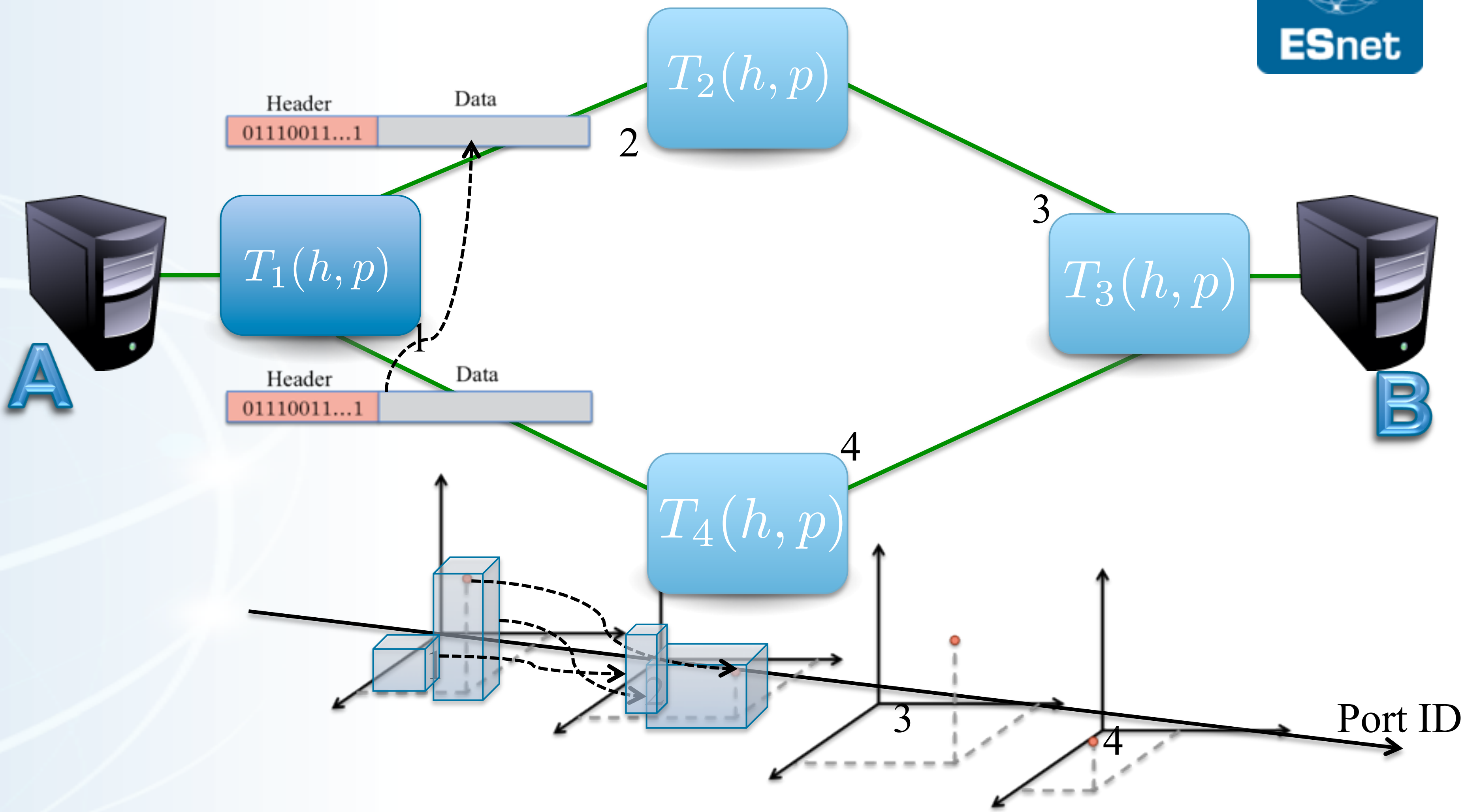
Slide courtesy Nick McKeown, ONS 2012

Header Space Analysis



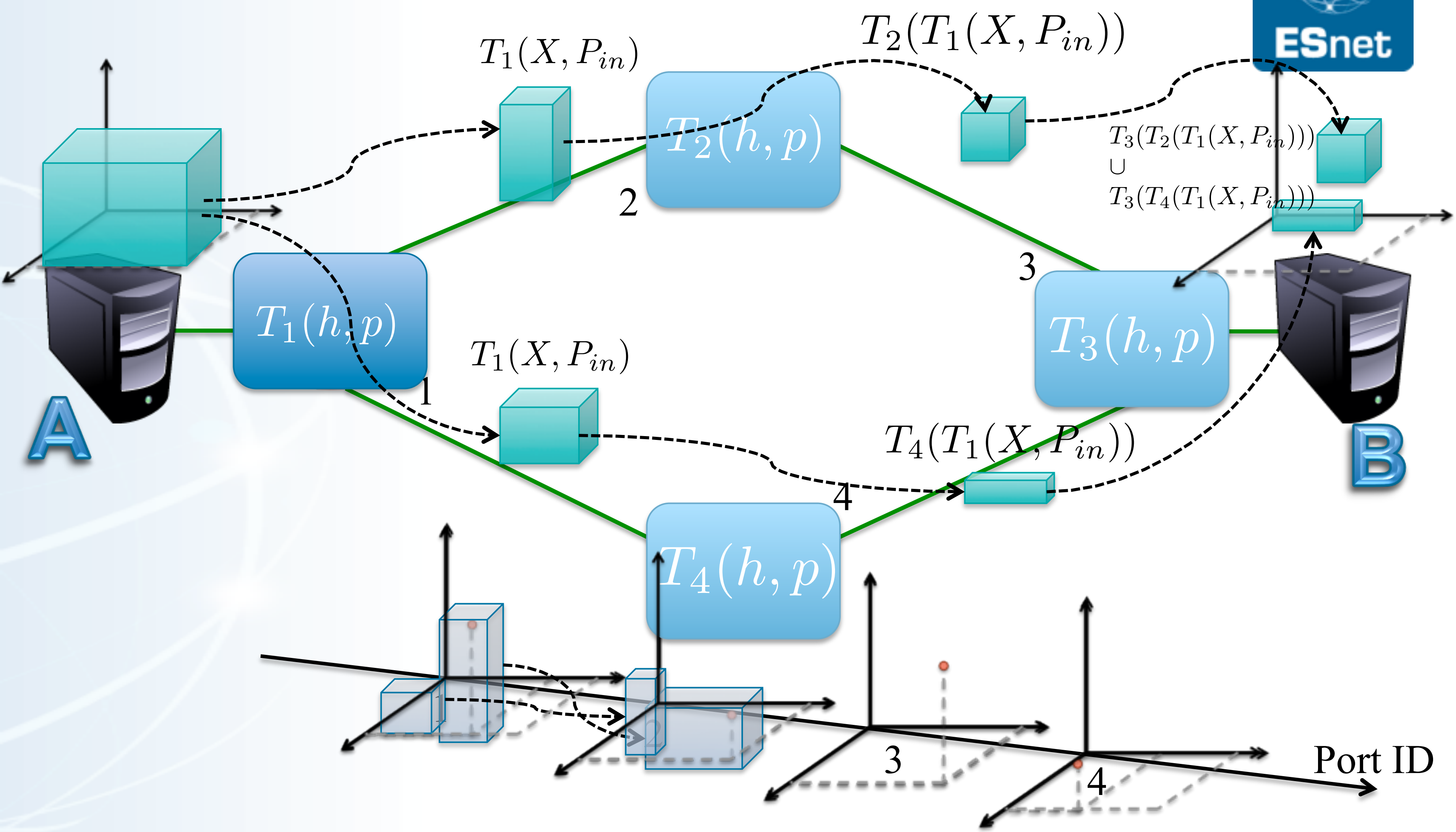
Slide courtesy Nick McKeown, ONS 2012

Header Space Analysis



Slide courtesy Nick McKeown, ONS 2012

Can A talk to B?



Header Space Analysis

Consequences

1. Finds <u>all</u> packets from A that can reach B
2. Find loops, regardless of protocol or layer
3. Can prove that two groups are isolated
4. Protocol Independent

Proves if network adheres to policy

Works on existing networks and SDNs

Slide courtesy Nick McKeown, ONS 2012

Stanford Backbone

Hassell tool

1. Reads Cisco IOS Configuration
2. Checks reachability, loops and isolation
3. 10 mins for Stanford Backbone
4. Easily made parallel: 1 sec is feasible

Hassell is available for free, for you to run

Slide courtesy Nick McKeown, ONS 2012

Summary, Opinions and Looking Forward

Software-Defined Networking is the “operative” word

- OpenFlow is a tool to implement the larger vision

SDN is still a tool

- Fine-granular customizability through programmability (application-engaged) are the ultimate goals
- Complexity is being tackled by software-tools to debug network problems deterministically

Common question: this is nothing new

- Likely true, why isn’t something new happening?
- MPLS started as IETF discussions 15 year after TCP/IP
- First, existing problems will likely be solved
- Enough momentum to take it over the hump (funding and attention)

These are my personal opinions and do not necessarily represent the position of DOE

Summary, Opinions and Looking Forward (2)

Research in using OpenFlow for SDN is a fertile area

- Lots of different solutions, programming languages and abstractions will be created before something settles
- ONF seems to have a strong role to play, how will non-members interact is an open-question

Network abstractions is going to be a key research topic

- The POSIX of networking still needs to get created
- It will take time to evolve

A programming language for networking primitives will be sorely needed

- Will have a lot of churn till network abstractions settle
- Exciting time for network engineers, though additional skillsets may require to be learnt.

These are my personal opinions and do not necessarily represent the position of DOE

Summary, Opinions and Looking Forward (3)

Separation of control and forwarding, with application programmability, finally creates a flexible network virtualization model

Common protocol (OF) enables software-solutions that “can” be truly multi-vendor

- Impact of vendor-specific extensions unclear

Central pre-planning of protection and restoration enabled

- Carriers do that today with EROs, easier here.
- Will the behavior and training evolve in large organizations?

Controller technology is still nascent (at least what is publicly known)

- Rich area for development

These are my personal opinions and do not necessarily represent the position of DOE

Summary, Opinions and Looking Forward (4)

The business model for OF is not research or cost savings through white-label boxes any more

- Solutions that you depend on 24x7 need to be well supported and likely from the same vendor

- Debugging, finger pointing etc can be an issue, but a lot of testing can happen before deploying.
  - Organizational discipline will need to be developed for this code.

Early on, hardware will likely support multi-functionality based on switch partitioning

These are my personal opinions and do not necessarily represent the position of DOE

Inder Monga

Imonga @ es.net

Thank You!